# An Enhanced One-round Pairing-based Tripartite Authenticated Key Agreement Protocol

Meng-Hui Lim*, Sanggon Lee**, Youngho Park***, Hoonjae Lee**

*Department of Ubiquitous IT, Graduate school of Design & IT, Dongseo University, Busan 617-716, Korea
meng17121983@yahoo.com
**Department of Information & Communication, Dongseo University, Busan 617-716, Korea
{nok60, hjlee}@dongseo.ac.kr
***School of Electronics and Electrical Engineering, Sangju National University, Sangju-Si, Gyeongsangbuk-do 742-711, Korea
yhpark@sangju.ac.kr

**Abstract.** A tripartite authenticated key agreement protocol is generally designed to accommodate the need of three specific entities in communicating over an open network with a shared secret key, which is used to preserve data confidentiality and integrity. Since Joux proposed the first pairing-based one-round tripartite key agreement protocol in 2000, numerous authenticated protocols have been proposed after then. However, most of them have turned out to be flawed due to their inability in achieving some desirable security attributes. In 2005, Lin-Li had identified the weaknesses of Shim's protocol and subsequently proposed their improved scheme by introducing an extra verification process. In this paper, we prove that Lin-Li's improved scheme remains insecure due to its susceptibility to the insider impersonation attack. Based on this, we propose an enhanced scheme which will not only conquer their defects, but also preserves the desired security attributes of a key agreement protocol.

## 1 Introduction

A *key agreement protocol* is defined as a mechanism in which a shared secret key, often known as *session key*, is derived by two or more protocol entities as a function of information contributed by each of these parties such that no single entity can predetermine the resulting value. Usually, this session key is established over a public network controlled by the adversaries and it may varies with every execution round (session) of the protocol. This secret key can subsequently be used to create a confidential communication channel among the entities.

Generally, a key agreement protocol is said to be *authenticated* if the protocol is able to ensure that the session key is known only to the intended entities in a protocol run. Without authentication, a key agreement protocol would probably turn out to be insecure since an adversary can easily intrude the scheme by using the man-in-the-middle attack as well as other related cryptographic attacks.

Wilson and Menezes [19, 20] have defined a number of desirable security attributes which are normally used to analyze key agreement protocols nowadays. These security attributes are described as follows:

**Known session key security.** A protocol is considered to be *known session key secure* if it remains achieving its goal in the face of an adversary who has learned some previous session keys.

**(Perfect) forward secrecy.** A protocol enjoys *forward secrecy* if the secrecy of the previous session keys is not affected when the long term private keys of one or more entities are compromised. *Perfect forward secrecy* refers to the scenario when the long term private keys of all the participating entities are compromised.

**Key-Compromise Impersonation Resilience.** Suppose that $A$'s long term private key is disclosed. Obviously an adversary who knows this value can now impersonate $A$ since it is precisely the value which identifies $A$. We say that a protocol is *key-compromise impersonation resilient* if this loss will not enable an adversary to masquerade as other legitimate entities to $A$ as well or obtain other entities' secret key.

**Unknown Key-Share Resilience.** In an unknown key-share attack, an adversary convinces a group of entities that they share a key with the adversary whereas in fact, the key is shared between the group and another party. This situation can be exploited in a number of ways by the adversary when the key is subsequently used to provide encryption of integrity.

**Key Control Resilience.** It should not be possible for any of the participants (or an adversary) to compel the session key to a preselected value or predict the value of the session key.

Over the years, countless key agreement protocols have been proposed. However, most of them have been proven to be insecure [1, 2, 3, 8, 11, 12, 14, 15] due to their inability in achieving all these desirable security attributes. In 2000, Joux [8] had proposed the first one-round pairing-based tripartite Diffie-Hellman key agreement protocol. However, Shim [15] had pointed out that Joux's protocol does not authenticate the communicating entities and therefore, it is susceptible to the man-in-the-middle attack. Furthermore, Shim had proposed an improved scheme which employs the public key infrastructure to overcome the security flaw in Joux's protocol and he claimed that the improved protocol is able to withstand the man-in-the-middle attack. Unfortunately in 2005, Lin-Li [10] had identified the weaknesses of Shim's improved scheme and subsequently proved its vulnerability to the insider impersonation attack as well as the key-compromise impersonation attack. In addition, Lin-Li had proposed their enhanced scheme by introducing an extra verification process in order to authenticate the communicating parties and they claimed that their enhanced scheme is secure and efficient. However, we discover that the extra verification process can be made ineffectual and their enhanced scheme is in fact breakable.

Hence, in this paper, we will prove that Lin-Li's one-round pairing-based tripartite authenticated key agreement protocol remains insecure due to its vulnerability to the insider impersonation attack. Based on this, we will propose our enhanced scheme which will not only conquer their defects, but also preserve the desired security attributes of a key agreement protocol. The structure of this paper is organized as follows. In the next section, we will illustrate some basic properties of modified Weil

pairings and some Diffie-Hellman assumptions. In section 3, we will review Lin-Li's one-round pairing-based tripartite authenticated key agreement protocol. Then, we will present our attacks on Lin-Li's scheme in section 4 and subsequently demonstrate our enhancements on their scheme as well as the associated security proofs in section 5. Last but not least, we will conclude this paper in Section 6.

# 2 Preliminaries

## 2.1 Modified Weil Pairing

Let $p$ be a prime number such that $p = 2 \pmod 3$ and

$$p = 6q - 1 \quad (1)$$

for some prime $q > 3$. Let $E[q]$ be a supersingular curve defined by

$$y^2 = x^3 + 1 \quad (2)$$

over $\mathbf{F}_p$. Let $P \in E/\mathbf{F}_p$ be a generator of the group of points with order

$$q = (p + 1)/6. \quad (3)$$

Let $\mu_q$ be a subgroup of $\mathbf{F}_{p^2}^*$ that contains all elements of order $q$. The Weil pairing on the curve $E/\ \mathbf{F}_{p^2}^*$ is a mapping $e$: $G_q$ x $G_q \rightarrow \mu_q$. Hence, we define the modified Weil pairing:

$$\hat{e}: G_q \text{ x } G_q \rightarrow \mu_q, \quad (4)$$

$$\hat{e}(P,Q) = e(P, \psi(Q)), \quad (5)$$

where $\psi(x, y) = (\zeta x, y)$, $1 \neq \zeta\ \mathbf{F}_{p^2}^*$ is a solution of $x^3 - 1 = 0 \pmod p$ and $G_q$ is the group of points with order $q$. The modified Weil pairing then satisfies the following properties:

a) **Bilinear**:

$$\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab} = \hat{e}(abP, Q) \quad (6)$$

for any $P, Q \in E[q]$ and $a, b \in Z_q^*$.

b) **Alternative:**

$$\hat{e}(P, Q) = \hat{e}(Q, P)^{-1}. \quad (7)$$

c) **Non-degenerate**: There exists a point $P \in G_q$ where $\hat{e}(P, P) \neq 1$.

d) **Polynomial-time computable**: There is an efficient algorithm to compute $\hat{e}(P,Q)$ for any $P, Q \in G_q$.

A bilinear map which satisfies all these properties above is known as *admissible bilinear*. It is noted that the modified Weil pairing associated with the supersingular elliptic curves or abelian varieties, can create such bilinear maps.

### 2.2 Diffie-Hellman Assumptions

Now, we can describe some hard cryptographic problems:

**Bilinear Diffie-Hellman Problem (BDHP)**. Let $\mathbf{G_1}$, $\mathbf{G_2}$ be two groups of prime order $q$ ($\mathbf{G_1}$ is an additive group and $\mathbf{G_2}$ is a multiplicative group). Let $P$ be a generator of $\mathbf{G_1}$. Given a quadruple $(P, aP, bP, cP)$ with $a, b, c \in Z_q^*$, compute $\hat{e}(P, P)^{abc} \in \mathbf{G_2}$. An algorithm $\alpha$ is deemed to have an advantage $\varepsilon$ in solving the BDHP in $(\mathbf{G_1}, \mathbf{G_2}, \hat{e})$ based on the random choices of $a, b, c$ in $Z_q^*$ and the internal random operation of $\alpha$ if

$$\Pr[\alpha((P, aP, bP, cP)) = \hat{e}(P, P)^{abc}] \geq \varepsilon. \quad (8)$$

**Discrete Logarithm Problem (DLP).** Given two groups of elements $P$ and $Q$, such that

$$Q = nP. \quad (9)$$

Find the integer $n$ whenever such an integer exists.

Throughout this paper, we assume that BDHP is a hard computational problem such that there is no polynomial time algorithm to solve BDHP and DLP with non-negligible probability.

## 3 Review of Lin-Li's Scheme

***Setup*:**

Suppose that three protocol principals $A$, $B$ and $C$ wish to communicate with each other by agreeing on a common session key. The public domain parameters ($p$, $q$, $E$, $P$, $\hat{e}$, $H$) are made common to all entities, where $H$: $Z \rightarrow Z$ is a predefined collision-free one-way hash function. Assume that the static public keys are exchanged via certificates. $Cert_A$ denotes $A$'s public-key certificate, containing his static public key

$$Y_A = aP \quad (10)$$

(where $a$ is $A$'s static private key) which uniquely identifies $A$ (such as $A$'s name and address), and a certification authority CA's signature over this information. Similarly, $Cert_B$ and $Cert_C$ are the certificates for $B$ and $C$ respectively, with

$$Y_B = bP \quad (11)$$

and

$$Y_C = cP \quad (12)$$

as their static public keys, where $b$ and $c$ are the long-term private keys selected by $B$ and $C$ respectively.

***Message Exchange*:**

Suppose that $x$, $y$ and $z$ are the ephemeral private keys chosen by $A$, $B$ and $C$ respectively in a communication round. Then, the message broadcast process can be accomplished as follows:

$A \rightarrow B, C$:

$$T_A = x \cdot (aP),\ Cert_A \tag{13}$$

$$m_A = H(ax) \tag{14}$$

$$s_A = (ax)^{-1}(m_A + a) \bmod q \tag{15}$$

$B \rightarrow A, C$:

$$T_B = y \cdot (bP),\ Cert_B \tag{16}$$

$$m_B = H(by) \tag{17}$$

$$s_B = (by)^{-1}(m_B + b) \bmod q \tag{18}$$

$C \rightarrow A, B$:

$$T_C = z \cdot (cP),\ Cert_C \tag{19}$$

$$m_C = H(cz) \tag{20}$$

$$s_C = (cz)^{-1}(m_C + c) \bmod q \tag{21}$$

***Message Verification*:**

$$t_A = s_A^{-1} \bmod q \tag{22}$$

$$u_A = (t_A m_A) \bmod q \tag{23}$$

$$t_B = s_B^{-1} \bmod q \tag{24}$$

$$u_B = (t_B m_B) \bmod q \tag{25}$$

$$t_C = s_C^{-1} \bmod q \tag{26}$$

$$u_C = (t_C m_C) \bmod q \tag{27}$$

$$u_A \cdot P + t_A \cdot Y_A \stackrel{?}{=} T_A \tag{28}$$

$$u_B \cdot P + t_B \cdot Y_B \stackrel{?}{=} T_B \tag{29}$$

$$u_C \cdot P + t_C \cdot Y_C \stackrel{?}{=} T_C \tag{30}$$

***Key Generation:***

$$K_A = \hat{e}\,(Y_B + T_B,\, Y_C + T_C)^{a + ax} = \hat{e}\,(P, P)^{(a + ax)\,(b + by)\,(c + cz)} \quad (\mathbf{31})$$

$$K_B = \hat{e}\,(Y_A + T_A,\, Y_C + T_C)^{b + by} = \hat{e}\,(P, P)^{(a + ax)\,(b + by)\,(c + cz)} \quad (\mathbf{32})$$

$$K_C = \hat{e}\,(Y_A + T_A,\, Y_B + T_B)^{c + cz} = \hat{e}\,(P, P)^{(a + ax)\,(b + by)\,(c + cz)} \quad (\mathbf{33})$$

a) $A$ computes Eqs. (24), (25), (26) and (27), and verifies whether Eqs. (29) and (30) hold simultaneously. If the verification process is successful, $A$ computes the session key by using Eq. (31).
b) $B$ computes Eqs. (22), (23), (26) and (27), and verifies whether Eqs. (28) and (30) hold simultaneously. If the verification process is successful, $B$ computes the session key by using Eq. (32).
c) $C$ computes Eqs. (22), (23), (24) and (25), and verifies whether Eqs. (28) and (29) hold simultaneously. If the verification process is successful, $C$ computes the session key by using Eq. (33).

Therefore, the shared session key can then be obtained as

$$K = kdf(K_A \| A \| B \| C) = kdf(K_B \| A \| B \| C) = kdf(K_C \| A \| B \| C), \quad (\mathbf{34})$$

where *kdf* is denoted as the key derivation function.

# 4 Our Attacks

Suppose that $W$ is the entity whose message is going to be authenticated, where $W \in \{A, B, C\}$ in this tripartite authentication protocol. It is crucial to note that there is no authentication provided for $T_W$ in the verification process. As a result, an adversary who impersonates $W$ can simply broadcast his desired value of $T_W$ in order to be accepted by the other legitimate entities in a protocol run. Hence in this section, we will demonstrate how this minor flaw can be exploited in launching the insider impersonation attack against their authenticated key agreement protocol.

## 4.1 Verification Process Infiltration

Assume that an adversary $E$, who wishes to impersonate entity $W$ in a protocol run, has obtained $Cert_W$ previously. Then, $E$ can launch his attack by carrying out the following procedures:

***Message Exchange*:**
First, $E$ selects two random numbers $j$, $n \in Z_q^*$, and computes

$$m_W' = H(n), \quad (\mathbf{35})$$

$$T_W' = (jm_W')P + jY_W, Cert_W \quad (36)$$

Then $E$ defines

$$s_W' = j^{-1} \quad (37)$$

and broadcasts his messages $T_W'$, $m_W'$ and $s_W'$ to the other protocol entities.

***Message verification***:
The legitimate entities will now compute

$$t_W = (s_W')^{-1} \bmod q = j \bmod q, \quad (38)$$

$$u_W = t_W m_W' \bmod q = jm_W' \bmod q, \quad (39)$$

and verify

$$u_W P + t_W Y_W = (jm_W')P + jY_W = T_W'. \quad (40)$$

Based on this, Eq. (40) will always be authenticated successfully regardless of the forged value of $T_W'$. As we have defined earlier in Section 2, DLP is hard. Hence, no single protocol entity would be able to suspect anything about $T_W'$. In other words, the protocol participants will mistakenly believe that entity $W$ is trying to communicate with them. In short, the verification process can easily be penetrated with a mere employment of Eqs. (35), (36) and (37). Once the authentication process has been made ineffectual, this authentication protocol can be proved insecure effortlessly due to its apparent susceptibility to the insider impersonation attack which we will demonstrate them in the next subsection.

### 4.2 Insider Impersonation Attack

In a two-party's authentication protocol, the adversary who impersonates the communicating parties would probably be an *outsider*. However, in the $n$-party's case where $n \geq 3$, the adversary who impersonates the communicating parties might be a legal entity of the communicating group, known as an *insider* and this kind of impersonation attack is called the *insider impersonation attack*. The consequence of this attack may be disastrous as the impersonated party might be a referee or an online escrow agent.

In Lin-Li's tripartite authentication scheme, suppose that $B$ is the insider impersonation attacker who wishes to fool $A$ by masquerading as $C$ to communicate with $A$. Assume that $B$ has obtained $Cert_C$ previously and $C$ has no knowledge about this communication run.

Based on these assumptions, $B$ then initiates a key agreement protocol with $A$. At the same time, $B$ also plays another role as $B_C$ ($B$ masquerading as $C$) and participates in this communication round. $B_C$ initially defines $m_C'$ $T_C'$ and $s_C'$ by using Eqs. (35), (36) and (37) respectively and broadcast them to $A$. The insider impersonation attack algorithm can be illustrated as follows:

$$A \rightarrow B, B_C: \{T_A, m_A, s_A, Cert_A\}$$
$$B \rightarrow A, B_C: \{T_B, m_B, s_B, Cert_B\}$$
$$B_C \rightarrow A, B: \{T_C', m_C', s_C', Cert_C\}$$

As proven in Section 4.1, $A$ will accept $T_C'$ since $A$ will eventually verify that Eq. (40) holds. Subsequently, the new session key can be computed by entities $A$, $B$ and $B_C$ as follows:

$$K_A = \hat{e}\,(Y_B + T_B,\, Y_C + T_C')^{a + ax} = \hat{e}\,(P, P)^{(a + ax)\,(b + by)\,j(m_c' + c)} \quad (41)$$

$$K_B = \hat{e}\,(Y_A + T_A,\, Y_C + T_C')^{b + by} = \hat{e}\,(P, P)^{(a + ax)\,(b + by)\,j(m_c' + c)} \quad (42)$$

$$K_{Bc} = \hat{e}\,(Y_A + T_A,\, Y_C + T_C')^{b + by} = \hat{e}\,(P, P)^{(a + ax)\,(b + by)\,j(m_c' + c)} \quad (43)$$

Since $B$ is able to derive the new session key $K_B$ and $K_{Bc}$ by using merely his long term private key $b$ and his ephemeral private key $y$, $B$ therefore has successfully fooled $A$ that $C$ has participated in a protocol run but in fact, C did not. Hence, we may conclude that Lin-Li's protocol is vulnerable to the insider impersonation attack.

# 5 Our Enhancement Scheme

As we have noticed in the previous section, Lin-Li's scheme has fallen into the insider impersonation attack mainly due to their failure in authenticating the value of $T_W$ in the verification process. In this section, we propose an enhanced one-round tripartite authenticated key agreement protocol based on Lin-Li's work in order to conquer their defects.

## 5.1 Protocol Improvement Description

**Setup & Message Exchange.** Our enhanced protocol has the same setup settings as Lin-Li's protocol. In the message exchange stage, $A$, $B$ and $C$ computes respectively

$$R_A = xP, \quad (44)$$

$$R_B = yP, \quad (45)$$

$$R_C = zP. \quad (46)$$

The message broadcast process is accomplished as follows:

$$A \rightarrow B, C: \{R_A, T_A, m_A, s_A, Cert_A\}$$
$$B \rightarrow A, C: \{R_B, T_B, m_B, s_B, Cert_B\}$$
$$C \rightarrow A, B: \{R_C, T_C, m_C, s_C, Cert_C\}$$

**Message Verification & Key Generation.** After receiving the messages from the other legitimate entities,

$$\hat{e}(T_A, P) = \hat{e}(R_A, Y_A) \quad (47)$$

$$\hat{e}(T_B, P) = \hat{e}(R_B, Y_B) \quad (48)$$

$$\hat{e}(T_C, P) = \hat{e}(R_C, Y_C) \quad (49)$$

a) $A$ authenticates $T_B$ and $T_C$ by verifying whether Eqs. (48) and (49) hold respectively. Then, $A$ computes Eqs. (24), (25), (26) and (27), and verifies whether Eqs. (29) and (30) hold simultaneously. If these 2 verification processes are successful, $A$ computes the session key by using Eq. (31). If any of the verification processes fails, $A$ then rejects.

b) $B$ authenticates $T_A$ and $T_C$ by verifying whether Eqs. (47) and (49) hold respectively. Then, $B$ computes Eqs. (22), (23), (26) and (27), and verifies whether Eqs. (28) and (30) hold simultaneously. If these 2 verification processes are successful, $B$ computes the session key by using Eq. (32). If any of the verification processes fails, $B$ then rejects.

c) $C$ authenticates $T_A$ and $T_B$ by verifying whether Eqs. (47) and (48) hold respectively. Then, $C$ computes Eqs. (22), (23), (24) and (25), and verifies whether Eqs. (28) and (29) hold simultaneously. If these 2 verification processes are successful, $C$ computes the session key by using Eq. (33). If any of the verification processes fails, $C$ then rejects.

### 5.2 Correctness

In our enhanced protocol, we have introduced an extra pairing operation in order to authenticate $T_A$, $T_B$, and $T_C$. Now, we demonstrate the correctness of Eqs. (47), (48) and (49) accordingly.

$$\hat{e}(T_A, P) = \hat{e}(axP, P) = \hat{e}(xP, aP) = \hat{e}(R_A, Y_A)$$

$$\hat{e}(T_B, P) = \hat{e}(byP, P) = \hat{e}(yP, bP) = \hat{e}(R_B, Y_B)$$

$$\hat{e}(T_C, P) = \hat{e}(czP, P) = \hat{e}(zP, cP) = \hat{e}(R_C, Y_C)$$

If any of $T_A$, $T_B$, and $T_C$ are forged by using Eq. (36), the legitimate entities would be able detect it since Eqs (47), (48) or (49) will not hold. Hence, a protocol entity would reject the session key if any of the verification process fails.

### 5.3 Protocol Security Analysis

In this subsection, we will examine our enhanced one-round tripartite authenticated key agreement protocol in order to ensure that the security attributes for a key agreement protocol are satisfied.

**Known session key security.** It is obvious that the session key of our protocol varies with every protocol run since it is established according to the values of the entities' ephemeral private keys ($x$, $y$ and $z$) in that particular session. Hence, the knowledge of past session keys would not allow the adversary to deduce any future session keys.

**Perfect forward secrecy.** Suppose that the entire long term private keys $a$, $b$ and $c$ have been compromised. In addition, assume that the adversary has also obtained some previously session keys established by the protocol entities. However, the adversary is unable to derive any other previously established session keys as shown in Eqs. (31) (32) and (33) since he does not possess the ephemeral private keys employed in that particular protocol run.

**Key-Compromise Impersonation Resilience.** Suppose that the long term private key $a$ has been compromised and the adversary wishes to impersonate $B$ in order to communicate with $A$. However, he is unable to forge $m_B$, $s_B$ and the session key $K_B$ since he does not know $b$. In addition, he is also unable to compute the session key $K_A$ since he does not know $x$. Since this protocol is symmetric, the same situation would results when the long term key $b$ or $c$ is compromised.

**Insider Impersonation Resilience.** Although an insider attacker, who wishes to impersonate $B$, could compute the session key, he could not forge $m_B$ and $s_B$ since he does not know $b$. Suppose that the attacker selects a random number, $y$', computes $T_B$ and $R_B$ by using Eqs. (45) and (16) correspondingly and forges $m_B$' and $s_B$'. Then, he initiates a protocol run and broadcast these values to the other intended entities. In the verification stage, the protocol entities would succeed in authenticating Eq. (48) but not Eq. (29). Even if the attacker employs the attack demonstrated in Section 4, the legitimate entities would have detected the counterfeit in verifying Eq. (48) and hence, the attacker would not be able to forge $T_B$ by using Eq. (36). In a nutshell, without knowing $B$'s long term private key, our protocol is immune to the insider impersonation attack.

**Key Control Resilience.** Apparently in our protocol, no single protocol participant could force the session key to a predetermined or predicted value since the session key of our protocol is derived by using the long term and ephemeral private keys of all the protocol participants.

## 6 Conclusion

In this paper, we have highlighted the flaws of Lin-Li's one-round authenticated tripartite key agreement protocol. By penetrating their verification process, we have further depicted the susceptibility of their scheme to the insider impersonation attack. To overcome this, we have demonstrated our improvements on their verification process by introducing an extra pairing operation in their authentication scheme. More significantly, we have carried out a thorough security analysis in order to scrutinize our enhanced scheme heuristically. Hence, we conclude that our enhanced tripartite authenticated key agreement protocol has been proven to be secure against various malicious attacks, while preserving the desired security attributes of a key agreement protocol.

# 7 Reference


1. S. S. Al-Riyami and K. G. Paterson, “Tripartite Authenticated Key Agreement Protocols from Pairings”, Cryptology ePrint Archive: Report, (035)(2002).
2. C. Boyd, W. Mao, K. G. Paterson. “Deniable Authenticated Key Establishment for Internet Protocols”, 11th International Workshop on Security Protocols, Cambridge (UK), April 2003.
3. Z. H. Cheng, L. Vasiu, and R. Comley, “Pairing-based One-round Tripartite Key Agreement Protocols”, Cryptology ePrint Archive, Report (079)(2004).
4. H. Y. Chien, “Comments: Insider Attack on Cheng et al’s Pairing-based Tripartite Key Agreement Protocols”, Cryptology ePrint Archive: Report, (013)(2005).
5. H. Y. Chien and R. Y. Lin, “An Improved Tripartite Authenticated Key Agreement Protocol Based on Weil Pairing”, Int. J. Appl. Sci. Eng., 2005. 3, 1.
6. J. S. Chou, Y. L. Chen, M. D. Yang, “Weaknesses of the Boyd-Mao Deniable Authenticated key Establishment for Internet Protocols”, Cryptology ePrint Archive: Report, (451)(2005).
7. J. S. Chou, C. H. Lin, C. H. Chiu, “Weakness of Shim’s New ID-based Tripartite Multiple-key Agreement Protocol”, Cryptology ePrint Archive: Report, (457)(2005).
8. A. Joux, “A One-round Protocol for Tripartite Diffie-Hellman”, Proceedings of the 4th International Algorithmic Number Theory Symposium (ANTS-IV), LNCS 1838, July, 2000, pp.385-394.
9. M. H. Lim, S. G. Lee, Y. H. Park, H. J. Lee, “An Enhanced ID-based Deniable Authentication Protocol on Pairings”, Cryptology ePrint Archive: Report, (113)(2007).
10. C. H. Lin, H. H. Li, “Secure One-Round Tripartite Authenticated Key Agreement Protocol from Weil Pairing”, Proceedings of the 19th International Conference on Advanced Information Networking and Applications (AINA 2005), pp.135-138.
11. D. Nalla, “ID-based Tripartite Key Agreement with Signatures", Cryptology ePrint Archive: Report, (144)(2003).
12. D. Nalla and K. C. Reddy, “ID-based tripartite Authenticated Key Agreement Protocols from pairings", Cryptology ePrint Archive: Report, (004)(2003).
13. K. Shim, “Cryptanalysis of Al-Riyami-Paterson's Authenticated Three Party Key Agreement Protocols", Cryptology ePrint Archive: Report, (122)(2003).
14. K. Shim, “Efficient ID-based Authenticated Key Agreement Protocol based on Weil Pairing”, Electronics Letters, Vol. 39, no. 8, April, 2003, pp.653-654.
15. K. Shim, “Efficient One-round Tripartite Authenticated Key Agreement Protocol from Weil Pairing”, Electronics Letters, Vol. 39, no. 2, January, 2003, pp.208-209.
16. H. M. Sun and B. T. Hsieh, “Security Analysis of Shim’s Authenticated Key Agreement Protocols from Pairings”, Cryptology ePrint Archive: Report, (113)(2003).
17. R. Tso, T. Okamoto, T. Takagi, E. Okamoto, “An ID-based Non-Interactive Tripartite Key Agreement Protocol with *K*-Resilience”, Communications and Computer Networks 2005, pp. 38-42.
18. Y. Xun, “Efficient ID-based Key Agreement from the Weil Pairing”, Electronics Letters, Vol. 39, no. 8, January, 2003, pp.206-208.
19. S. B. Wilson, and A. Menezes, “Authenticated Diffie-Hellman key agreement protocols”, Proceedings of the 5th Annual Workshop on Selected Areas in Cryptography (SAC ’98), LNCS, (1999) (339-361).
20. S. B. Wilson, D. Johnson and A. Menezes, “Key Agreement Protocols and their Security Analysis”, Proceedings of the 6th IMA International Conference on Cryptography and Coding, Vol. 1355, LNCS, pp. 339-361. Springer-Verlag, 1998.

21. F. G. Zhang, S. L. Liu, K. J. Kim, “ID-based One Round Authenticated Tripartite Key Agreement Protocol with Pairings”, Cryptology ePrint Archive: Report, (122)(2002).